PortTerm

User Manual

Version.1.5

14 June 2004

Content

1. Getting Started

This chapter introduces the PortTerm software and describes how to install the software on your computer.

1.1. About PortTerm

PortTerm is a software application designed to provide developers with easy access to serial ports under Mac OS X. It allows reading and writing serial ports via a GUI.

1.2. Registration

PortTerm is shareware. If you use it, you should register this product. The license fee is $20US. Payments can be made via PayPal to vkulesh@oakland.edu. Alternatively, international customers can pay the fee using an International Money Order. A fast and convenient way of sending a money order is via www.payingfast.com. The US customers can pay by money order or check. The Michigan residents must include a 6% sales tax. All payment must be in US funds. The mailing address is:

Victor Kulesh
41575 Wessel Drive
Sterling Heights, MI 48313
USA

When sending a payment, include a valid e-mail address. If you wish to obtain a copy of PortTerm on CD, include an additional $5 (within the United States only).

1.3. Legal

The PortTerm software and the accompanying materials are provided "AS IS" without warranty of any kind. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE AUTHOR(S) HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. The entire risk as to the results and performance of this software is assumed by you. If the software is defective, you, and not Victor Kulesh, assume the entire cost of all necessary servicing, repairs and corrections.

1.4. System Requirements

PortTerm is a native Mac OS X application and will work on any Macintosh computer running OS 10.2 or later. Most Macintosh computers come without physical serial ports and thus a USB to Serial adapter is required. An example of such adapter is Keyspan's Serial Adapter. PortTerm relies on the drivers provided by the manufacturer of

the adapter. If the drivers for a specific adapter are not available then PortTerm can not access that device.

1.5. Installing PortTerm

PortTerm is distributed as a disk image file called PortTerm.dmg. Use the “Disk Copy” application located in Applications/Utilities to mount the disk image on your computer’s desktop. Drag and drop the PortTerm application to any folder on your hard drive and you are done with the installation.

2. Using PortTerm

This chapter describes the functionality provided by PortTerm and how to use its features.

2.1. Selecting the serial port

Immediately after launching, PortTerm scans the /dev directory for installed serial device drivers. Those devices drivers that are recognized as serial are added to the Current Port menu. Use this menu to select the serial port that you are going to use. In the example below the serial adapter from Keyspan named USA19QI1814P1.1 is selected as the current serial port. The current port selection is saved and will be the default serial port, next time PortTerm is launched.

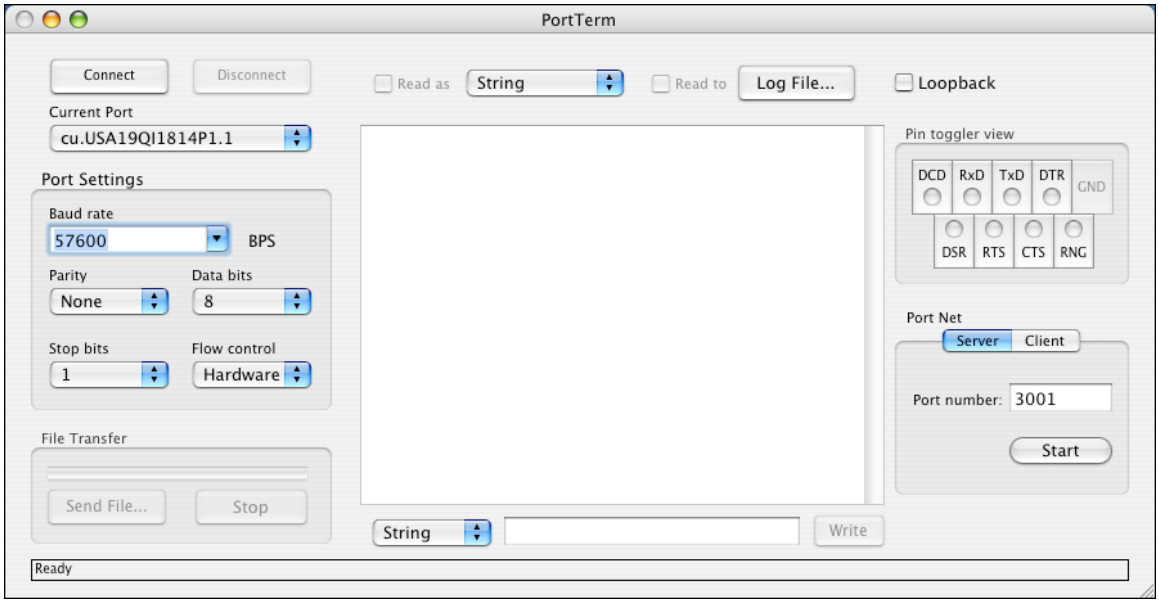

Figure 1. PortTerm serial port selection menu

If your serial port adapter is not recognized by PortTerm, see the Frequently Asked Questions chapter of this document.

2.2. Configuring the serial port

After selecting the serial port it must be configured by selecting the options supported by your external device. Choose the baud rate, parity, number of data bits and stop bits. Not all serial adapters support flow control and custom baud rates. Check with the adapter's manufacturer if the feature is implemented.

Most external devices only work with the same settings for both input and output; therefore, PortTerm uses the same setting for both input and output.

Once the serial port is configured, click the "Connect" button to open the serial port. PortTerm is ready to read and/or write your data.

2.3. Reading from the serial port

PortTerm allows reading raw data from the current port. The data can be either displayed in a text view in two different formats or written to a file. The first method is convenient for development work and is recommended for small amounts of data. The second method is preferred when large amounts of data are logged.

2.3.1. Reading Data into a text view

In order to start reading the data from the serial port, choose the format in which the data should be presented. Currently two options are available: hexadecimal and characters (String). String is the default view format. Select the "Read as" checkbox to begin reading the data from the port. The data will be displayed in the window. The amount of data that can be read and displayed in the window is limited. If you use PortTerm for logging a lot of data from external devices, then it is recommended to use the Log to File feature of PortTerm.

2.3.2. Reading Data directly to a file

PortTerm allows data logging to a file. To enable data logging, use the "Log File…" button to specify the name of the file, where the data should be saved. If an existing file is selected, the new data will be appended at the end of the existing data. Then use the "Read to" checkbox to start/stop data logging.

2.4. Writing to the serial port

PortTerm allows writing data to the current serial port using a variety of formats. The default format for writing data is a sequence of characters, where each character is one byte. Other writing formats include decimal and hexadecimal.

2.4.3. Writing character strings

Character strings or a sequence of characters is the default format for writing data to the serial port with PortTerm. Select “String” from the write format pop-up menu, type the string and click the “Write”” button to send the sequence of characters to the port. The sequence of characters is interpreted by PortTerm as a sequence of bytes, which are written to the port without any extra processing.

Figure 2. Writing in string format

2.4.4. Writing decimal values

To write a number to the serial port, select “Decimal” from the Write Format pop-up menu, type the number and click the “Write” button.

Figure 3. Writing in decimal format

The decimal value is treated as a signed integer. The values in the range [-128, 127] will be written to the port using one byte, the values in the range [-32768, 32767] will be written using two bytes, all other will be written using four bytes.

2.4.5. Writing hexadecimal values

To write a number to the serial port in hexadecimal form, select “Hex” from the Write Format pop-up menu, type the value in hexadecimal format and click “Write” button.

Figure 4. Writing in hexadecimal format

For values in the range [0x00, 0xFF] one byte of data will be written, for values [0x0100, 0xFFFF] two bytes of data will be used and for all others four bytes of data.

When writing numeric values only one number at a time can be written to the serial port. Any sequence of digits is interpreted as one number.

2.4.6. Writing a sequence of hexadecimal values

To write a sequence of numbers in hex format to the serial port, select “Hex Seq” from the Write Format pop-up menu, type the values in hexadecimal format separating them by space and click “Write” button. Note that only white space can be used as a separator! Each value will be treated as a one byte number.

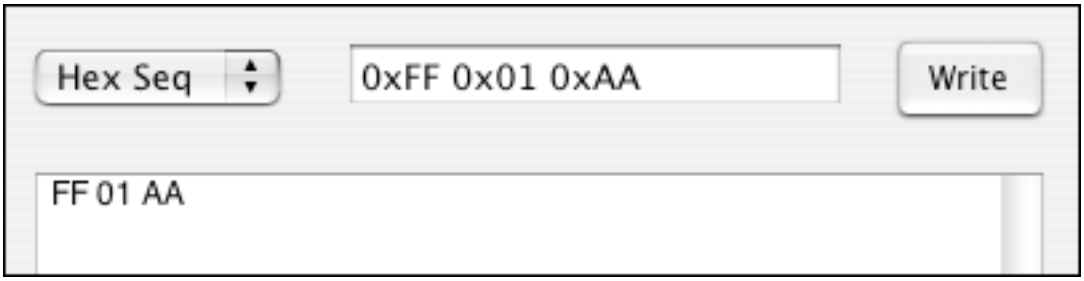

Figure 5. Writing a sequence of hexadecimal values

The picture above shows how to send a sequence of three hex values: “0xFF 0x01 0xAA” to the serial port.

2.4.7. Writing data from a file

PortTerm allows writing data to the serial port from a data file. Click “File send…” button and choose the data file. Alternatively, from the “File” menu, select the “Send…” menu item and choose the data file. Each byte of data from the file will be written without any preprocessing. The progress bar will show the amount of data written to the port. The writing process can be interrupted at any time by clicking on the “Stop” button next to the progress bar.

2.4.8. Sending AT Commands

To send an AT command, connect to a device capable of handling AT commands, choose “AT cmd” from the write format menu, select “String” from the read format menu and click on the “Read as” checkbox. Type your AT command in the writing field and click the “Write” button or “Return” on your keyboard. The AT command will be sent along with the ‘\r’ at the end of the command. An example of sending “AT” to a built-in modem is shown below.

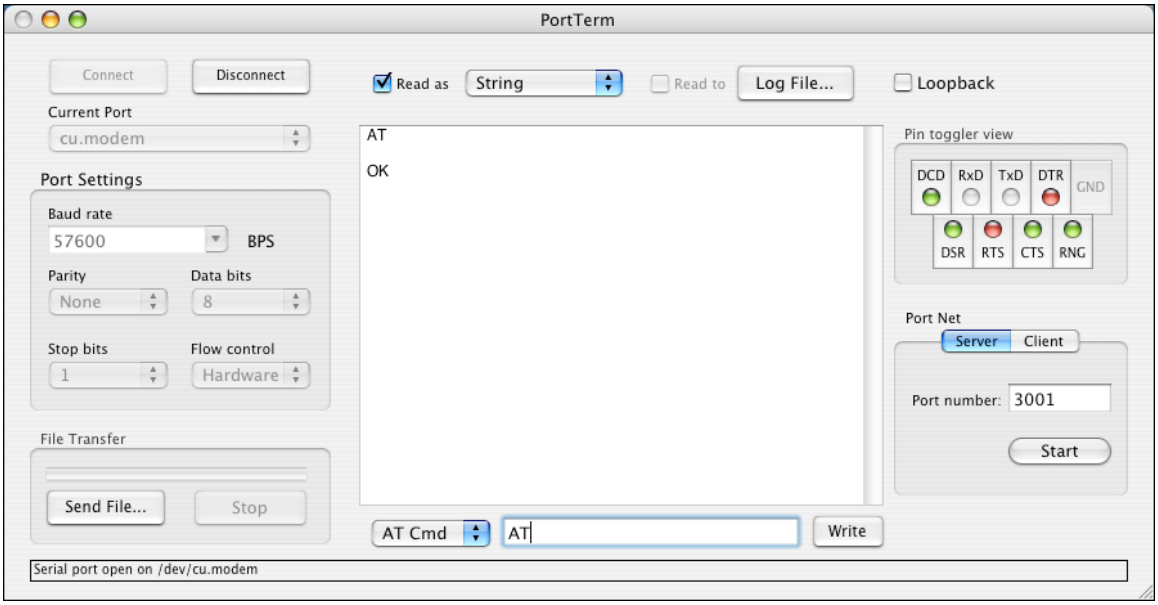

Figure 6. Writing AT commands

An “AT” written to cu.modem was echoed back and then the modem replied with “OK".

2.5. PortNet

PortNet is built-in server that allows access to the serial port to clients over the TCP/IP network. Essentially, this feature will turn any device with a serial port into Internet ready device.

2.5.9. Using PortNet

Connect to your serial device as you would normally do with PortTerm. Select “Server” tab from the PortNet view. Choose the port number for PortNet server to listen for incoming connections and click “Start” button.

Use a client application such as telnet to connect to PortTerm. For example when using telnet on the same computer one would type: **telnet 127.0.0.1 3001** to connect to PortNet on the same computer on port 3001. When connecting from a remote computer you have to know both the IP address and the port number. Here is an example on how one would connect on a LAN: **telnet 192.168.0.100 3001.** This assumes that PortTerm is running on a computer with an IP address of 192.168.0.100 and port number 3001.

All data that the client sends will be written to the serial port and any data coming from the serial port will be forwarded back to the client over TCP/IP. When the client closes the connection PortNet server will be ready to accept a new connection.

In order to stop the server, click “Stop” button. This will close the server and if there is an open connection from the client, it will be terminated immediately. At this point you can disconnect from the serial device.

2.6. PinToggler

PinToggler is built-in tool that allows PortTerm to control the individual lines of the serial port.

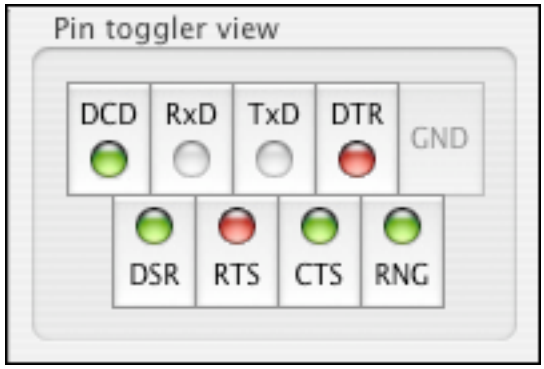

Figure 7. PinToggler View

The output lines (DTR and RTS) can be toggled by clicking on the corresponding buttons. If hardware flow control is used, then only DTR line can be toggled from the Pin Toggler. The green LED indicates the negative voltage on the line, while the red LEDs indicate the positive voltage on the line. The input lines (DCD, DSR, CTS and RNG) are monitored and their states are reflected by the color of the LED. When there is no input signal on the line, the corresponding LED will be green. When a signal is present on the input line, the LED will automatically change color to red.

2.6.10. TxD (Transmit Data)

TxD is the data output from the computer to the device. It is used to send data bits to the device from the computer. This line is not controlled by PinToggler.

2.6.11. DTR (Data Terminal Ready)

DTR signal is generated by the computer and tells the device attached to the serial port that the computer is ready (space) or not ready (mark). Usually DTR is enabled when the serial port is opened successfully. When the DTR line has negative (mark) voltage, the DTR LED is green. When the DTR line has positive (space) voltage, the DTR LED is red. Clicking on the DTR button will toggle the line.

2.6.12. RTS (Request To Send)

RTS signal is set to space by the computer informing the device attached to the serial port that the computer has more data ready to be sent. When the RTS line has negative (mark) voltage, the RTS LED is green. When the RTS line has positive (space) voltage, the RTS LED is red. Clicking the RTS button will toggle the line. If hardware flow control is used, the RTS line can NOT be controlled from Pin Toggler.

2.6.13. RxD (Receive Data)

RxD is the data input from the device to the computer. It is used to receive data from the device. The RxD LED will flash when data is received. This line is not controlled by PinToggler.

2.6.14. DCD (Data Carrier Detect)

The DCD signal is generated by the device attached to the serial port. The space on the DCD line indicates that the device is currently online. DCD is not always used.

2.6.15. CTS (Clear To Send)

The CTS signal is generated by the device attached to the serial port. The space on the CTS line indicates that the device is ready to send data. CTS is not always used or supported by the manufacturers of the devices.

2.6.16. RNG (Ring indicator)

The RNG also known as RI signal is generated by the device attached to the serial port and tells the compute that the device is wants to open a connection. RNG is not always used or supported by the manufacturers of the devices.

2.6.17. DSR (Data Set Ready)

The DSR signal is generated by the device attached to the serial port and tells the compute that the device has data ready to be sent to the computer. DSR is not always used or supported by the manufacturers of the devices.

3. AppleScript Support

This chapter shows how to make use of PortTerm's AppleScripting capabilities and gives examples of some basic scripts.

3.1. Supported Commands

PortTerm supports the following AppleScript commands:

- connect
- configure
- write
- read
- disconnect

The "connect" command must be executed before any other command is issued to PortTerm. "Connect" requires the name of the port as a parameter. Here is how to properly connect to a serial port from AppleScript:

```
connect to port "cu.modem"
– where "cu.modem" is the name of the port you are connecting to
```

The next step is configuring the serial port. The "configure" command with appropriate parameters must be used. "Configure" takes the following parameters:
baud rate, parity, data bits and stop bits. Possible values for baud rate are: "300", "600", "1200", …, "115200". Possible value for parity are: "none", "even" and "odd". Possible value for data bits are "8", "7", "6" and "5". The values for stop bits are either "1" or "2". If any of the parameters are not provided a default value of the parameter will be used that are "1200" baud for the baud rate, "none" for parity, "8" for data bits and "1" for stop bits. Here is how one would use the "configure" command:

```
configure baud rate "2400" parity "none"
```

Once the port is configured you can write to and read from it. Use the "write" command to write data to the serial port. The "write" command requires the data parameter and an optional format parameter. If the format parameter is not used then "write" writes the data to the serial port treating the data as a sequence of bytes. If the format is used then the data is written according to the format specified. Currently only "atcm" value (AT Command) for format is supported; using format "atcm" will append '\r' at the end of the data sequence. If you need to write hex values to the serial port, use AppleScript to build a string, where each byte is a hex value and then write it as a string. Here is an example of using the "write" command:

```
write data "AT" format "atcm"
```

Reading the data from the serial port is a two-step process. Firstly, issue the "read" command. The "read" command will read the data (if any) and place it into a variable called iData. The iData will contain the data and should be interpreted as a string by your script. Here is an example of reading the data from the serial port and the displaying it in a dialog.

```
read
display dialog (iData as string)
```

If you write something to the serial port and the device sends a response, keep in mind that it make take some time for the data to appear in the serial port's buffer, so make sure you wait a little bit before issuing the read command.

Finally, to disconnect from the serial port, use "disconnect" command.

Here is a full-fledged AppleScript example that writes an "AT" command to a serial modem and then reads a response.

```
tell application "PortTerm"
	activate
	connect to port "cu.modem"
	configure baud rate "2400" parity "none"
	write data "AT" format "atcm"
	delay 4 --give the modem some time to respond
	read -- read the response
	display dialog (iData as string)
	disconnect
end tell
```

More examples will be posted at the following web site: http://home.comcast.net/~vkulesh. If you have any script that you are willing to share with others, please email them to vkulesh@oakland.edu and they will be posted as well.

4. Frequently Asked Questions

This chapter describes common problems that developers encounter when using PortTerm and provides answers to some of the Frequently Asked Questions.

4.1. PortTerm does not see my device.

PortTerm relies on the serial port drivers provided by the manufacturer of the serial adapter. Not all adapters have drivers for Mac OS X. Make sure yours does. If your adapter supports OS X, make sure the drivers are installed. Open the Terminal and type the following commands:

% cd /dev

% ls

Look for files with the following name: cu.xxxxxxx where xxxxxxx will be determined by your particular device. Consult the adapter's software manual for exact names of the device drivers.

4.2. Can I get the source code for PortTerm?

No. However, I will help registered users of the software with code samples to access serial ports.